An iterative algorithm for finite element analysis

Farid Laouafa, P. Royis

An iterative algorithm for finite element analysis

F. Laouafa* & P. Royis**

* Institut National de l'Environnement Industriel et des RISques
Parc Technologique ALATA, BP 2
60550 Verneuil en Halatte, France

** Ecole Nationale des Travaux Publics de l'Etat
Département Génie Civil et Bâtiment (URA CNRS 1652)
rue Maurice Audin, F-69518 Vaulx-en-Velin Cedex, France

Abstract

In this paper we state in a new form the algebraic problem arising from the one-field displacement finite element method. The displacement approach, in this discrete form, can be considered as the dual approach (force or equilibrium) with subsidiary constraints. This approach dissociates the non-linear operator to the linear ones and their sizes are linear functions of integration rule which is of interest in the case of reduced integration. This new form of the problem leads to an inexpensive improvement of F.E.M. computations, which acts at local, elementary and global levels. We demonstrate the numerical performances of this approach which is independent of the mesh structure. By using the GMRES algorithm, we build for nonsymmetric problems, a new algorithm based upon the discretized field of strain. The new algorithms proposed are more closer to the mechanical problem than the classical ones because all fields appear during the resolution process. The sizes of the different operators arising in these new forms are linear functions of integration rule, which is of great interest in the case of reduced integration.

KEY WORDS: finite element method; mixed element; GMRES algorithm; strain algorithm.

1. INTRODUCTION

In structural and solid mechanics analysis, the most frequently used finite element method is probably the one based on the one field variational formulation involving the displacement field $u(x,t)$ [10,17,29] which leads classically to the well known algebraic problem :

$$\begin{cases} \text{find } U \in \mathbb{R}^N \text{ such that :} \\ \mathbf{K} \cdot U = F \qquad \mathbf{K} \in \mathbb{R}^{N \times N}, \quad F \in \mathbb{R}^N \end{cases} \tag{1}$$

where ${}^tU = \{u_1, u_2, \cdots, u_N\}$ is the vector of the unknown nodal displacements, in which N denotes the total number of degrees of freedom. The vector F in the right-hand side of (1) is the vector of the nodal forces, whereas $\mathbf{K}$ denotes the $N \times N$ (linear or not, symmetric or not) stiffness matrix. Classically, both $\mathbf{K}$ and F are built from the assembly of elementary contributions,

$$\mathbf{K} = \sum_{e=1}^{nel} \mathbf{K}_e, \quad \mathbf{K}_e \in \mathbb{R}^{n_e \times n_e} \qquad \text{and} \qquad F = \sum_{e=1}^{nel} F_e, \quad F_e \in \mathbb{R}^{n_e}$$

where n_e denotes the element number of degrees of freedom. In rocks mechanics or in soil mechanics as well as for material involving concrete the non-normality rule of the constitutive law leads to a non-symmetric stiffness matrices. The range of such mechanical problems involving non-symmetries is rather large and needs special attention when solving in boundary values problems. The building of the elementary stiffness matrices $\mathbf{K}_e$ and therefore that of the global stiffness matrix $\mathbf{K}$ leads to an increasing computation cost and to the loss of the local mechanical informations, in particular those relating to the constitutive law, and then requires post-analysis to recover these local features.

Within the same mathematical framework we propose a new method for building and solving the algebraic problems arising from the one-field (displacement) FEM. This approach, called UDA for Unstructured Displacement Approach, can be considered as an IBI (Integration point By Integration point) method. In this method we state the problem (1) by using three operators $\hat{\mathbf{B}}$, $\hat{\mathbf{H}}$ and $\hat{\mathbf{I}}$. The first one is the global strain-displacement operator, the second $\hat{\mathbf{H}}$ is the 'rheological' operator the components of which are the local (i.e. by integration point) constitutive relationships. The third $\hat{\mathbf{I}}$ is the one associated with the weak equilibrium and numerical integration rule. These three operators remain distinct and uncoupled during the whole resolution of the problem. In order to decrease the computational time linked with the updating and assembling process of the Hessian matrix, Axelsson et al. proposed an interesting and efficient factorization of the stiffness matrix (see ref. 1 and 2 for details). Our paper focus only on non-symmetric problems of solid continua and on the contribution of UDA to some iterative methods.

These methods take a great advantage of the numerical integration rules which makes possible to dissociate the kinetic, rheological (constitutive law) and static local features of the disctretized variational problem.

The structure of these operators remains mesh (h and p) independent. This approach takes advantage of the new form of the problem and of the features of the GMRES algorithm[24,25], both of them providing then substancial improvements to the one-field displacement FEM. The memory storage required by the UDA method is less than that required by the classical sky line[8] method, the sparse matrix method or by other methods like those based on an element by element approach[8]. Thus, the computational times is greatly reduced. The above characteristics increase greatly when reduced integration is used. This new important feature is due to the fact that the parameters linked with the integration rule are dissociated during the whole resolution of the algebraic problem. Many mechanical problems require adaptive mesh refinement. In this framework the new structure of the problem increase the efficieny of the method.

In the UDA method the algebraic problem coming from the continuous mechanical problem is stated in a new form similar to the one arising from the stress-displacement two-fields mixed finite element method. In its algebraic form the displacement approach can be then considered as the dual approach (force or equilibrium) with subsidiary constraints. A unification of both one-field and two-fields methods is also proposed. From an algebraic point of view it completes the unification proposed by Malkus and Hughes[14] and Zienkiewicz and Nakazawa[28].

We mention that the algorithms presented in this paper are also available for a real stress-displacement two-fields mixed finite element analysis.

2. CLASSICAL APPROACHS

The general mechanical framework of our study lies in problems characterized by a solid continuum Ω subject to external body forces $f(x,t)$ and surface forces $g(x,t)$ prescribed on a part Γ_2 of its boundary Γ. The part $\Gamma_1 = \Gamma - \Gamma_2$ of the boundary of the solid is fixed. Let us denote as $u(x,t)$ the displacement field and as $\boldsymbol{\sigma}(x,t)$ the second order tensor of the Cauchy stresses. We assume that Ω is a materially simple continuum. We suppose also small transformations, so

that the strain tensor $\boldsymbol{\epsilon}(u(x,t))$ (symmetric part of the gradient of the displacement) is linear. Eventually we restrict our analysis to quasi-static problems.

The constitutive law can be described by incrementally non-linear equations[3,20] of the form :

$$\dot{\epsilon} = \mathbf{J}(d,\chi)\cdot\dot{\sigma}, \quad d = \frac{\dot{\sigma}}{\|\dot{\sigma}\|}, \quad (\dot{\epsilon},\dot{\sigma}) \in \mathbb{R}^{n_{\sigma n}} \times \mathbb{R}^{n_{\sigma n}}, \quad \mathbf{J}(d,\chi) \in \mathbb{R}^{n_{\sigma n}\times n_{\sigma n}} \tag{2}$$

where $\dot{\boldsymbol{\epsilon}}$ is the strain rate vector (i.e. the vector built from the components of the strain rate tensor $\dot{\epsilon}$), $\dot{\boldsymbol{\sigma}}$ the stress rate vector of size $n_{\sigma n}$, and where χ denotes the set of memory parameters. More generally the constitutive law can be an associated or non-associated elasto-plastic one written as follows:

$$\sigma = \mathbf{H}(\chi)\cdot\epsilon, \quad (\epsilon,\sigma) \in \mathbb{R}^{n_{\sigma n}} \times \mathbb{R}^{n_{\sigma n}}, \quad \mathbf{H}(\chi) \in \mathbb{R}^{n_{\sigma n}\times n_{\sigma n}} \tag{3}$$

In this paper one can be consider equally well both of relationships (2) and (3). We only focus our attention on non-symmetric problem such as those encountered in soils, in a rocks mechanics or in physical problems involving non-symmetric constitutive law.

The well known one-field (displacement) variational formulation of this mechanical problem is classically stated as,

$$(\mathcal{PV}) \begin{cases} \text{Find } u \in \mathbf{V} = \{v \in (\mathbf{H}^1_{(\Omega)})^3, \quad v \mid \Gamma_1 = 0\} \quad \text{such that :} \\ \displaystyle\int_\Omega (\mathbf{H} : \boldsymbol{\epsilon}(u)) : \boldsymbol{\epsilon}(v)\, d\Omega = \int_\Omega f\cdot v\, d\Omega + \int_{\Gamma_2} g\cdot v\, d\Gamma \quad \forall v \in \mathbf{V} \end{cases} \tag{4}$$

and leads, after a space discretization, to the algebraic problem :

$$(\mathcal{S}) \begin{cases} \text{find } U \in \mathbb{R}^N \text{ such that :} \\ \mathbf{K}\cdot U = F \qquad \mathbf{K} \in \mathbb{R}^{N\times N}, \quad F \in \mathbb{R}^N \end{cases} \tag{5}$$

The space discretization leads to the following finite sum involving the contributions of the finite elements Ω_e [10,29]

$$\mathbf{K} = \sum_{e=1}^{nel} \mathbf{K}_e \quad \text{with} \quad \mathbf{K}_e = \int_{\Omega_e} {}^t\mathbf{B}_e\, \mathbf{H}\, \mathbf{B}_e\, d\Omega \tag{6}$$

The numerical evaluation of $\mathbf{K}_e$ is usually made by using a Gaussian quadrature method, so that the elementary stiffness matrices can be constructed as :

$$\mathbf{K}_e = \sum_{k=1}^{npi_e} {}^t\mathbf{B}_e(x_k)\, \mathbf{H}(x_k)\, \mathbf{B}_e(x_k)\, \omega_k\, j_k \tag{7}$$

with :

$$\begin{cases} npi_e & \text{: number of integration points used for the finite element } e, \\ \omega_k & \text{: the weight of the transformation at the integration point } k, \\ j_k & \text{: the value of the Jacobian of the transformation at the integration point } k. \end{cases}$$

and the local displacement-strain relationship :

$$\epsilon_e^k = \mathbf{B}_e(x_k)\cdot U_e, \quad \epsilon_e^k \in \mathbb{R}^{n_{\sigma n}}, \quad U_e \in \mathbb{R}^{ne} \quad \mathbf{B}_e(x_k) \in \mathbb{R}^{n_{\sigma n}\times n_e} \tag{8}$$

The actual iterative methods[5,9,22] for solving (5) work generaly with the global stiffness matrix $\mathbf{K}$ or with its elementary contributions $\mathbf{K}_e$. The method described below takes advantage of the numerical evaluation of the element stiffness matrices.

3. NEW FORM OF THE FINITE ELEMENT PROBLEM

The principal features of the new form of the finite element problem are linked to the numerical integration of the elementary stiffness matrices. Using a Gaussian quadrature method one can rewrite [12,13] problem (1) in the form :

$$(\mathcal{S}) \begin{cases} \text{Find } U \in \mathbb{R}^N \text{ such that :} \\ \left({}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}}\right) \cdot U = F \end{cases} \tag{9}$$

We can also formulate the one-field problem in an equivalent stress-displacement two-fields mixed form [13]:

$$(\mathcal{S}) \begin{cases} \text{Find } (\underline{\sigma}, U) \in \mathbb{R}^{N_\sigma} \times \mathbb{R}^N \text{ such that :} \\ \begin{pmatrix} \hat{\mathbf{H}}^{-1} & -\hat{\mathbf{B}} \\ {}^t\hat{\mathbf{B}}\hat{\mathbf{I}} & \mathbf{0} \end{pmatrix} \left| \begin{matrix} \underline{\sigma} \\ U \end{matrix} \right| = \left| \begin{matrix} 0 \\ F \end{matrix} \right| \end{cases} \tag{10}$$

where $\hat{\mathbf{B}}$ is the $N_\sigma \times N$ *Global* displacement-*Global* strain operator :

$$\underline{\epsilon} = \hat{\mathbf{B}} \cdot U, \qquad \hat{\mathbf{B}} \in \mathbb{R}^{N_\sigma \times N}, \quad \underline{\epsilon} \in \mathbb{R}^{N_\sigma}, \quad U \in \mathbb{R}^N \tag{11}$$

with the following definition of $\underline{\epsilon}$:

$${}^t\underline{\epsilon} = \left({}^t\epsilon_1, {}^t\epsilon_2, \cdots, {}^t\epsilon_{nel}\right) \quad \text{with} \quad {}^t\epsilon_e = \left({}^t\epsilon_e^1, \cdots, {}^t\epsilon_e^{npie}\right), \quad \epsilon_e^k \in \mathbb{R}^{n_{\sigma n}}$$

The stress size N_σ is equal to the sum $\sum_{e=1}^{nel} n_{\sigma n}\ npi_e$.

The matrix $\hat{\mathbf{H}}$ is the $N_\sigma \times N_\sigma$ *Global* constitutive matrix :

$$\underline{\sigma} = \hat{\mathbf{H}} \cdot \underline{\epsilon}, \qquad \hat{\mathbf{H}} \in \mathbb{R}^{N_\sigma \times N_\sigma} \quad \underline{\sigma} \in \mathbb{R}^{N_\sigma} \tag{12}$$

with the following definition of $\underline{\sigma}$:

$${}^t\underline{\sigma} = \left({}^t\sigma_1, {}^t\sigma_2, \cdots, {}^t\sigma_{nel}\right) \quad \text{with} \quad {}^t\sigma_e = \left({}^t\sigma_e^1, \cdots, {}^t\sigma_e^{npie}\right), \quad \sigma_e^k \in \mathbb{R}^{n_{\sigma n}}$$

Let us mention that $\hat{\mathbf{H}}$ is bloc-diagonal with respect to each integration point:

$$\hat{\mathbf{H}} = \begin{bmatrix} \mathbf{H}_1^1 & & \cdots & 0 \\ & \mathbf{H}_1^2 & & \\ \vdots & 0 & \ddots & \\ 0 & & & \mathbf{H}_{nel}^{npi_{nel}} \end{bmatrix} \quad \text{with} \quad \sigma_k^e = \mathbf{H}_k^e \cdot \epsilon_k^e, \qquad \mathbf{H}_k^e \in \mathbb{R}^{n_{\sigma n} \times n_{\sigma n}}$$

The diagonal matrix $\hat{\mathbf{I}} \in \mathbb{R}^{N_\sigma \times N_\sigma} = \sum_e \mathbf{I}e$ contains all informations in relation with the integration rules : the weights (ω) and the jacobian values (j) for all the integration points.

$$\mathbf{I}_e = \begin{bmatrix} (\omega j)_e^1 \mathbf{I}_d & \cdots & 0 \\ \vdots & \ddots & \\ 0 & & (\omega j)_e^{npi_e} \mathbf{I}_d \end{bmatrix} \quad \text{where } \mathbf{I}_d \text{ is the } n_{\sigma n} \times n_{\sigma n} \text{ identity matrix.}$$

The matrix $\hat{\mathbf{B}}$ is a sparse matrix, the components of which are the partial derivatives of the displacement interpolation functions. This operator is linear with respect to its argument U. It is of interest to remark that for non-linear problems arising, for instance, from elastoplasticity, this operator remains the same at each step of loading. So it is constant during all the iterations of the iterative process of resolution. Moreover, this operator remains constant for the

whole finite element analysis of the problem if one does not proceed to the update of the node coordinates. This can be done without increasing error of evaluation under the assumption of small transformations. So only one computation is to be done to form $\hat{\mathbf{B}}$, whatever the number of loading steps, of non-linear iterations and the of iterations linked with the solving iterative process.

Lets us now recapitulate the sizes of the different matrices encountered in this section and the corresponding storage required.

1- Matrix $\hat{\mathbf{B}}$:

a. size : $N_\sigma \times N$,

b. storage : $\beta_1 \sum_e npi_e \cdot (n_{\sigma n} \times n_e)$.

2- Matrix $\hat{\mathbf{H}}$:

a. size : $N_\sigma \times N_\sigma$,

b. storage : $\beta_2 \sum_e npi_e \cdot (n_{\sigma n} \times n_{\sigma n})$.

3- Matrix $\hat{\mathbf{I}}$:

a. size : $N_\sigma \times N_\sigma$,

b. storage : $\sum_e npi_e$.

The factors β_1 and β_2 are reduction factors depending respectively of the kind of problem (plane stress or plane strain problem, axisymmetric problem, etc...) and of the symmetry properties of the constitutive relations. Their possible values are as follows :

$$\beta_1 = \begin{cases} \dfrac{3}{2\, n_{\sigma n}} & \text{For axisymmetric problems without torsion} \\ \dfrac{1}{n_{\sigma n}} & \text{otherwise} \end{cases}$$

$$\beta_2 = \begin{cases} \dfrac{1}{2} + \dfrac{1}{2\, n_{\sigma n}} & \text{if } \mathbf{H} \text{ is symmetric} \\ 1 & \text{otherwise} \end{cases}$$

When solving a boundary value problem, one have to check the quality of the solution by some energy criteria or stress equilibrium[27]. When adaptive (h and p) refinement is performed, the modification of the mesh lead, in our method, only to an addition and/or modification of some finite elements operator. That do not destruct the structure of the algebraic problem and do not require the computation and modification of the other operators, and moreover the implementation in finite element code is very easy. This is a interesting feature when adaptive refinement is used intensively.

4. ONE AND EQUIVALENT TWO-FIELDS ALGEBRAIC PROBLEM

To obtain problem (9) we start from the discretized weak form of equilibrium relations which can be stated in term of stress unknowns and written as follows [12,13] :

$$^t\hat{\mathbf{B}}\,\hat{\mathbf{I}} \cdot \underline{\sigma} = F \tag{13}$$

By taking into account relation (12),

$$\underline{\sigma} = \hat{\mathbf{H}} \cdot \underline{\epsilon}, \qquad \underline{\sigma} \in \mathbb{R}^{N_\sigma}, \ \underline{\epsilon} \in \mathbb{R}^{N_\sigma}$$

we can express the weak form (13) in terms of strain :

$$ {}^t\hat{\mathbf{B}}\, \hat{\mathbf{I}}\, \hat{\mathbf{H}} \cdot \underline{\epsilon} = F \tag{14}$$

By using relation (11) we can now write the algebraic problem (1) in its two equivalent forms :

$$\text{Form I} \quad (\mathcal{S})^{uda} \begin{cases} \text{Find } U \in \mathbb{R}^N \text{ such that :} \\ \left({}^t\hat{\mathbf{B}}\, \hat{\mathbf{I}}\, \hat{\mathbf{H}}\, \hat{\mathbf{B}} \right) U = F \end{cases}$$

$$\text{Form II} \quad (\mathcal{S})^{uda} \begin{cases} \text{Find } (\underline{\sigma}, U) \in \mathbb{R}^{N_\sigma} \times \mathbb{R}^N \text{ such that :} \\ \begin{pmatrix} \hat{\mathbf{H}}^{-1} & -\hat{\mathbf{B}} \\ {}^t\hat{\mathbf{B}}\hat{\mathbf{I}} & \mathbf{0} \end{pmatrix} \begin{vmatrix} \underline{\sigma} \\ U \end{vmatrix} = \begin{vmatrix} 0 \\ F \end{vmatrix} \end{cases} \tag{15}$$

4.1. *Two-fields considerations.*

Let us consider the algebraic problem written in the form (II).

$$(\mathcal{S})^{uda} \begin{cases} \text{Find } (\underline{\sigma}, U) \in \mathbb{R}^{N_\sigma} \times \mathbb{R}^N \text{ such that :} \\ \begin{pmatrix} \hat{\mathbf{H}}^{-1} & -\hat{\mathbf{B}} \\ {}^t\hat{\mathbf{B}}\hat{\mathbf{I}} & \mathbf{0} \end{pmatrix} \begin{vmatrix} \underline{\sigma} \\ U \end{vmatrix} = \begin{vmatrix} 0 \\ F \end{vmatrix} \end{cases} \tag{16}$$

and let us multiply the first equation of the system (16) by $\hat{\mathbf{I}}$. We obtain

$$\hat{\mathbf{H}}^{-1} \cdot \underline{\sigma} - \hat{\mathbf{B}} \cdot U = 0 \quad \Leftrightarrow \quad \hat{\mathbf{I}}\, \hat{\mathbf{H}}^{-1} \cdot \underline{\sigma} - \hat{\mathbf{I}}\, \hat{\mathbf{B}} \cdot U = 0$$

Let us now consider the second equation in (16). Since $\hat{\mathbf{I}}$ is a diagonal matrix this relation can be written as :

$$ {}^t\hat{\mathbf{B}}\, \hat{\mathbf{I}} \cdot \underline{\sigma} = F \quad \Leftrightarrow \quad {}^t\left(\hat{\mathbf{I}}\, \hat{\mathbf{B}} \right) \cdot \underline{\sigma} = F$$

Then the system (16) takes the following form :

$$(\mathcal{S})^{uda} \begin{cases} \text{Find } (\underline{\sigma}, U) \in \mathbb{R}^{N_\sigma} \times \mathbb{R}^N \text{ such that :} \\ \begin{pmatrix} \hat{\mathbf{I}}\, \hat{\mathbf{H}}^{-1} & -\left(\hat{\mathbf{I}}\, \hat{\mathbf{B}} \right) \\ {}^t\left(\hat{\mathbf{I}}\, \hat{\mathbf{B}} \right) & \mathbf{0} \end{pmatrix} \begin{vmatrix} \underline{\sigma} \\ U \end{vmatrix} = \begin{vmatrix} 0 \\ F \end{vmatrix} \end{cases} \tag{17}$$

The algebraic problem given by the system (17) arises from the displacement one-field approach. However, we recognize an algebraic problem coming from the discretization of a stress-displacement two-fields mixed variational formulation. This last formulation can be stated as follows[12,13,21] :

$$(\mathcal{PV})^m \begin{cases} \text{Find} \quad (\boldsymbol{\sigma}, u) \in \mathbf{S} \times \mathbf{V} \quad \text{such that :} \\ \displaystyle\int_\Omega \left(\mathbf{H}^{-1} : \boldsymbol{\sigma} \right) : s \, d\Omega - \int_\Omega s : \boldsymbol{\epsilon}(v) \, d\Omega = 0 \qquad \forall s \in \mathbf{S} \\ \displaystyle\int_\Omega \boldsymbol{\sigma} : \boldsymbol{\epsilon}(w) \, d\Omega = \int_\Omega f \cdot w \, d\Omega + \int_{\Gamma_2} g \cdot w \, d\Gamma \quad \forall w \in \mathbf{V} \end{cases} \tag{18}$$

where $\mathbf{S}$ is the stresses trial functions space ; $\mathbf{S} = \{s_{ij} \in L^2_{(\Omega)} \quad \forall i,j \in \{1,2,3\}^2 \quad \text{and} \quad s_{ij} = s_{ji}\}$.

The algebraic problem (17) can be also obtained from (18) under some conditions relating to the finite element spaces of displacements $\mathbf{V}_h \subset \mathbf{V}$ and of stresses $\mathbf{S}_h \subset \mathbf{S}$. These conditions, which are independent of the position of the stress nodes inside the elements, generalizes the ones done by Malkus and Hughes[14] and Zienkiewicz[28,29]. They can be stated as follows :

$$\begin{cases} \hat{\mathbf{I}}\,\hat{\mathbf{H}}^{-1} & = \sum_{e=1}^{nel} \int_{\Omega_e} {}^t\boldsymbol{\psi}_e\, \mathbf{H}^{-1}\, \boldsymbol{\psi}_e \, d\Omega \\ \hat{\mathbf{I}}\,\hat{\mathbf{B}} & = \sum_{e=1}^{nel} \int_{\Omega_e} {}^t\boldsymbol{\psi}_e\, \mathbf{B}_e \, d\Omega \end{cases} \tag{19}$$

where $\boldsymbol{\psi}_e$ is the elementary interpolation matrix of the discontinuous stress field in the case of the two-fields mixed discretization.

$$\sigma(x) = \boldsymbol{\psi}_e(x) \cdot \sigma_e, \qquad \forall\, x \in \Omega_e, \quad \sigma_e : \text{elementary vector of stress components}$$

We have then established, under algebraic considerations, that a one field algebraic problem can always be expressed in its equivalent two-fields algebraic form. From a certain point of view the displacement one-field algebraic problem without subsidiary constraint is equivalent to its stress dual algebraic problem with subsidiary constraints. These constraints are the discretized weak form of equilibrium. We remark that they are not formulated in a local form, but in a weak one.

Theorem : *The stationary condition without subsidiary constraint of the discrete potential energy functional* $\Pi_p^d(U)$ ($U \in \mathbb{R}^N$ such that $u = 0$ on Γ_1),

$$\Pi_p^d(U) = \frac{1}{2}\, {}^tU \cdot \mathbf{K} \cdot U - {}^tU \cdot F$$

is equivalent to the stationary condition with subsidiary constraints of the discrete complementary energy functional $\Pi_c^d(\Sigma)$,

$$\Pi_c^d(\Sigma) = \frac{1}{2}\, {}^t\Sigma \cdot \hat{\mathbf{I}}\, \hat{\mathbf{H}}^{-1} \cdot \Sigma, \quad \Sigma \in \mathcal{K} \quad \mathcal{K} = \left\{\Sigma \in \mathbb{R}^{N_\sigma} \quad \text{such that} \quad {}^t\mathbf{B}\hat{\mathbf{I}} \cdot \Sigma = F\right\}$$

So, the equivalent discrete two-fields formulation is obtain by imposing the stationary condition of the discrete Lagrangian $\mathcal{L}^d(\Sigma, V)$, obtained by introducing the subsidiary constraint $\Sigma \in \mathcal{K}$

$$\mathcal{L}^d(\Sigma, V) = \Pi_c^d(\Sigma) - {}^tV \cdot \left({}^t\mathbf{B}\hat{\mathbf{I}} \cdot \Sigma - F\right)$$

The algebraic form (II) of the displacement finite element formulation is interesting for several reasons. The first of them is that for a certain class of non-linear constitutive equations[4,20] written as :

$$\epsilon(x) = \mathbf{J}(x) \cdot \sigma(x), \quad x \in \Omega.$$

the constitutive operator $\hat{\mathbf{J}}$ coming from $\mathbf{J}$ and built like $\hat{\mathbf{H}}$ does not need to be explicitly inverted. This constitutes a great difference and provide substantial advantages in comparison to the classical approach, because of the decreasing of calculations and of the decreasing of errors due to that inversion in the case of ill conditioned constitutive relations. The second reason, which holds also for the form (I) of the algebraic problem, is that the performances of the algorithms described below increase if one adopts reduced integration rules.

Considering the form (II) of the algebraic problem, we can easily give a condition for integration rule. Two levels of condition are required.

1- The global level, which ensures the injectivity of $\hat{\mathbf{B}}$:

$$\sum_{e=1}^{nel} npi_e \geq \frac{n_u - bcl}{n_{\sigma n}}$$

where bcl is the number of components of U prescribed.

2- The element level, which ensures the injectivity of $\mathbf{B}_e$:

$$npi_e \geq \frac{n_{u_e} - bcl_e}{n_{\sigma n}}$$

where bcl_e is the number of components of U_e prescribed. As established by Zienkiewicz[27,28] these conditions give some restrictions for reduced integration rule. As we can see by considering the algebraic forms (II) the reduced integration rule can be interpreted, from a two-fields mixed point of view, by the relaxation of the conditions of equilibrium. A physical interpretation can be done by considering the stress-displacement two-fields approach, and one can linked the LBB[2] condition to the integration rule.

5. FIRST COMPUTATIONAL ADVANTAGES

In this section we describe some features of the new approach proposed, which are independent of the method used to solve the linear or linearized problem. We give some numerical examples showing comparisons with the EBE method. In section 6 we give some real applications comparing our method and the sparse matrix method which used only nonzero entries.

5.1. *Storage allowed.*

The total storage required for $\hat{\mathbf{H}}$, $\hat{\mathbf{B}}$ and $\hat{\mathbf{I}}$ is :

$$S_t = \beta_2 \sum_e npi_e\, n_{\sigma n}^2 \;+\; \beta_1 \sum_e npi_e\, n_{\sigma n}\, n_e \;+\; \sum_e npi_e \tag{20}$$

Lets us call ρ_s the ratio between the storage required by $\mathbf{K}$ and that required by $\hat{\mathbf{B}}$, $\hat{\mathbf{H}}$ and $\hat{\mathbf{I}}$.

$$\rho_s = \frac{\text{storage required by } \mathbf{K}}{\text{storage required by } \hat{\mathbf{B}} \text{ , } \hat{\mathbf{H}} \text{ and } \hat{\mathbf{I}}}$$

In the frame of an Element By Element approach, the saving of storage increases with h and p refinement. Under the same assumption that all the needed operators are stored, The ratio ρ_s between the storages needed by EBE and UDA approach is :

$$\rho_s = \frac{nel\, n_e^2}{\beta_1 \sum_e^{nel} npi_e\, n_{\sigma n}\, n_e \;+\; \beta_2 \sum_e^{nel} npi_e\, n_{\sigma n}^2 \;+\; \sum_e^{nel} npi_e}$$

In the case of a same p refinement ($npi_e = npi, n_e = n\ \forall e$) we have :

$$\rho_s = \frac{n^2}{\beta_1\, npi\, n_{\sigma n}\, n \;+\; \beta_2\, npi\, n_{\sigma n}^2 \;+\; npi}$$

By using an exact integration rule for all the finite elements we have, for plane ($n_{\sigma n} = 3$) and 3-D problems ($n_{\sigma n} = 6$) :

2D-Finite element	npi	ρ_s ($\beta_1 = 1/n_{\sigma n}$, $\beta_2 = 1$)
Bilinear Lagrange	1×1	3.56
Linear triangle	1	2.25
Biquadratic Lagrange	2×2	2.89
Quadratic triangle	3	2.18
Bicubic Lagrange	3×3	2.71
Cubic triangle	6	2.22

Table 1. Values of ρ_s for some 2-dimensional elements.

3D-Finite element	npi_e	ρ_s ($\beta_1 = 1/n_{\sigma n}$, $\beta_2 = 1$)
Trilinear Lagrange	$1 \times 1 \times 1$	9.44
Linear tetrahedron	1	2.93
Triquadratic Lagrange	$2 \times 2 \times 2$	6.96
Quadratic tetrahedron	4	3.36
Tricubic Lagrange	$3 \times 3 \times 3$	5.99
Cubic tetrahedron	10	2.71

Table 2. Values of ρ_s for some 3-dimensional elements.

For EBE or for the classical approach using $\mathbf{K}$ the cost of storage does not depend on the number of integration points. So this cost remains the same if a reduced integration is used and thus whatever the methods employed. As we can see by considering equation (20), S_t is a function of the size of the various operators and of parameters β_1 and β_2. Therefore, a reduced integration leads to an important reduction of storage. The magnification m of ρ_s, which is a function of the number of reduced integration point npi_r is equal to :

$$m = \frac{npi}{npi_r} \tag{21}$$

The reduced integration rule is based on the order of the integrated polynomial. If npi is the quadrature rule used to integrate a polynomial function of degree n then npi_r is the one which integrates a polynomial function of order $n-1$. For instance we have, for the finite element of continua presented above,

2D-Finite element	npi_r	m
Biquadratic Lagrange	1	4
Quadratic triangle	1	3
Bicubic Lagrange	2×2	2.25
Cubic triangle	4	1.5

3D-Finite element	npi_r	m
Triquadratic Lagrange	1	8
Quadratic tetrahedron	1	4
Tricubic Lagrange	$2 \times 2 \times 2$	3.375
Cubic tetrahedron	4	2.5

Table 3. Magnification m of ρ_s for some 2-D and 3-D elements.

As we can see in tables 3 the magnification m of ρ_s is significant. This parameter is also equal to the ratio between the stress (or strain) size of the problems obtained by using an exact integration and that of problems corresponding to a reduced integration. This reduced integration leads then to an increasing speed of computation of the resolution process. This is not a feature of the classical approach.

5.2. *Computation time.*

The matrix-vector products in a EBE method is worth n_e^2 for one element. In UDA method, the cost matrix-vector products for one element is equal to the sum of :

$$\begin{aligned} \mathbf{B}_e \cdot U_e &= 2\, npi_e\, n_e \\ \mathbf{H}_e \cdot \epsilon_e &= npi_e\, n_{\sigma n}^2 \\ {}^t\mathbf{B}_e \cdot \sigma_e &= 2\, npi_e\, n_e \end{aligned}$$

Thus equal to $npi_e\,(4\,n_e + 9)$ for two dimensional problems and to $npi_e\,(6\,n_e + 36)$ for 3-D problems. Let us call ρ_t this cost obtain with an exact integration rule and ρ_{tr} the one obtained with reduced integration rule. We give in tables 4 and 5 the values of n_e^2, ρ_t and ρ_{tr} for the most used finite elements.

2D-Finite element	n_e	n_e^2	npi	ρ_t	ρ_{tr}
Bilinear Lagrange	8	64	1	41	41
Linear triangle	6	36	1	33	33
Biquadratic Lagrange	18	324	4	324	81
Quadratic triangle	12	144	3	171	57
Bicubic Lagrange	32	1024	9	1233	548
Cubic triangle	20	400	6	536	268

Table 4. Values of n_e^2, ρ_t and ρ_{tr} for some 2-dimensional elements.

3D-Finite element	n_e	n_e^2	npi	ρ_t	ρ_{tr}
Trilinear Lagrange	24	576	1	180	180
Linear tetrahedron	12	144	1	108	108
Triquadratic Lagrange	81	6561	8	4176	522
Quadratic tetrahedron	30	900	4	864	648
Tricubic Lagrange	192	36864	27	32076	9504
Cubic tetrahedron	48	2304	10	3240	1296

Table 5 . Values of n_e^2, ρ_t and ρ_{tr} for some 3-dimensional elements.

This subsection dealt with the reduction of the computation time induced by the consideration of operators $\hat{\mathbf{H}}$, $\hat{\mathbf{B}}$ and $\hat{\mathbf{I}}$ instead of $\mathbf{K}$ or $\mathbf{K}_e$, but not with the computation time required by the solver. These advantages relating to this last point are to be added to those described above, and to the post-processing step.

For a non-linear problem where the non-linearities are induced by the material behaviour, the only non-linear operator is the constitutive one $\hat{\mathbf{H}}$, since we assume small transformations. This leads to a unique computation of operators $\hat{\mathbf{B}}$ and $\hat{\mathbf{I}}$ for the global analysis of the mechanical problem.

Thus, with classical EBE or with $\mathbf{K}$ a non-linear analysis leads to update all of them at ecah iteration. Theses methods do not benefit of the fact that the matrices $\hat{\mathbf{B}}$ and $\hat{\mathbf{I}}$ remain constant, because the main operators $\mathbf{K}$ or $\mathbf{K}_e$ have to be formed for each iteration of the linearization process.

6. THE ITERATIVE ALGORITHMS

The various algorithms presented below can be used equally well for displacement one-field finite element analyses or for real stress-displacement two-fields mixed ones. For this last problems, we have only to substitute $\mathbf{A}$ to $\hat{\mathbf{I}}\,\hat{\mathbf{H}}^{-1}$ and $\mathbf{B}$ to $\hat{\mathbf{I}}\,\hat{\mathbf{B}}$, where $\mathbf{A}$ and $\mathbf{B}$ are defined by the following relations.

$$\begin{cases} \mathbf{A} = \sum_{e=1}^{nel} \int_{\Omega_e} {}^t\boldsymbol{\psi}_e\, \mathbf{H}^{-1}\, \boldsymbol{\psi}_e \, d\Omega \\ \mathbf{B} = \sum_{e=1}^{nel} \int_{\Omega_e} {}^t\boldsymbol{\psi}_e\, \mathbf{B}_e \, d\Omega \end{cases}$$

in which $\boldsymbol{\psi}_e$ denotes the interpolation functions of stress :

$\sigma(x) = \psi_e(x) \cdot \sigma_e, \quad \sigma_e$: elementary vector of stress components.

Lets us remind the reader that the problem modelled together with its equivalent form obtained after linearization at a given loading step are as follows:

$$(\mathcal{S})^{classical} \begin{cases} \text{Find } U \in \mathbb{R}^N \text{ such that :} \\ \mathbf{K} \cdot U = F \end{cases} \Leftrightarrow (\mathcal{S})^{uda} \begin{cases} \text{Find } U \in \mathbb{R}^N \text{ such that :} \\ \left({}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\hat{\mathbf{B}}\right) \cdot U = F \end{cases} \tag{22}$$

6.1 *GMRES method.*

Let $\|.\|$ and $(.,.)$ denote respectively the Euclidian norm and inner product. The iterative method used for the resolution of the non-symmetric algebraic problem (22) is the GMRES one [22]. This numerical method, which is a particular case of the Petrov-Galerkin method, is based upon a minimization of the Euclidian norm of the residual vectors r_m,

$$r_m = F - \mathbf{K} \cdot x_m$$

in a Krylov space $\mathcal{K}_m$ of dimension m,

$$\mathcal{K}_m = \mathcal{K}_m(\mathbf{K}) = \text{span}\left\{r_0, \mathbf{K} \cdot r_0, \mathbf{K}^2 \cdot r_0, \cdots, \mathbf{K}^{(m-1)} \cdot r_0\right\}.$$

Let us consider an approximate solution of the form $x_0 + z_m$, where x_0 is an initial guess and z_m belongs to the space $\mathcal{K}_m$. Then the above minimization problem can be expressed as follows [22] :

$$\min_{z_m \in \mathcal{K}_m} \|F - \mathbf{K} \cdot (x_0 + z_m)\| = \min_{z_m \in \mathcal{K}_m} \|r_0 - \mathbf{K} \cdot z_m\| \tag{23}$$

This algorithm uses an orthonormal basis $\boldsymbol{V}_m = [v_1, v_2, \cdots, v_m]$ of $\mathcal{K}_m$ which is built with the classical Arnoldi [22] procedure. If we take as the operator the stiffness matrix $\mathbf{K}$, the restarted GMRES(m) flowchart is given, in a synthetic form, by :

Initialization :

v_0 initial guess

$[\dagger]\ r_0 = F - \mathbf{K} \cdot v_0$

$$v_1 = \frac{r_0}{\|r_0\|}$$

Building of the orthonormal basis $\boldsymbol{V}_m$ of $\mathcal{K}_m$:

$$\left.\begin{array}{lll} h_{i,j} & = (\mathbf{K} \cdot v_j, v_i) \quad i = \{1, 2, \cdots, j\} & \\ \tilde{v}_{j+1} & = \mathbf{K} \cdot v_j - \displaystyle\sum_{i=1}^{j} h_{i,j} v_i & \\ h_{j+1,j} & = \|\tilde{v}_{j+1}\| & \\ v_{j+1} & = \dfrac{\tilde{v}_{j+1}}{\|\tilde{v}_{j+1}\|} = \dfrac{\tilde{v}_{j+1}}{h_{j+1,j}} & \end{array}\right\} j = \{1, 2, \cdots, m\}$$

Resolution of the reduced minimization problem :

$$v_m = v_0 + \boldsymbol{V}_m \cdot y, \quad \text{with } y \in \mathbb{R}^m \text{ minimizes } \|e - \boldsymbol{H}_m \cdot y\|$$

Convergence check :

$\|r_m = F - \mathbf{K} \cdot v_m\| \leq \varepsilon$ (userspecified tolerance)

If convergence check is not satisfied then affect v_m to v_0 and go to †.

In the algorithm above $\boldsymbol{H}_m$ is the $(m+1) \times (m)$ upper Hessenberg matrix :

$$\boldsymbol{H}_m = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,m} \\ \|\tilde{v}_2\| & h_{2,2} & \cdots & h_{2,m} \\ 0 & \|\tilde{v}_3\| & & \vdots \\ \vdots & & & \vdots \\ & & \|\tilde{v}_m\| & h_{m,m} \\ 0 & 0 & 0 & \|\tilde{v}_{m+1}\| \end{bmatrix}$$

and $e = {}^t(\|r_0\|, 0, \cdots, 0) \in \mathbb{R}^{m+1}$ is a basis vector. Let us denote this algorithm as GMRES-K. We shall now proceed to the substitution of $\mathbf{K}$ by the operators described in the previous sections.

6.2. *Substitution.*

The algorithm given above is written for an approach based on the stiffness matrix. If we now apply this algorithm to the dissociated form (I), it will be called GMRES-UDA and is as follows :

Flowchart of GMRES-UDA

v_0 initial guess

$[\dagger]\ r_0 = F - {}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_0$

$v_1 = \dfrac{r_0}{\|r_0\|}$

$$\left.\begin{aligned} h_{i,j} &= \left({}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_j, v_i\right) \quad i = \{1, 2, \cdots, j\} \\ \tilde{v}_{j+1} &= {}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_j - \sum_{i=1}^{j} h_{i,j} v_i \\ h_{j+1,j} &= \|\tilde{v}_{j+1}\| \\ v_{j+1} &= \frac{\tilde{v}_{j+1}}{\|\tilde{v}_{j+1}\|} = \frac{\tilde{v}_{j+1}}{h_{j+1,j}} \end{aligned}\right\} j = \{1, 2, \cdots, m\}$$

$v_m = v_0 + \boldsymbol{V}_m \cdot y$, with $y \in \mathbb{R}^m$ minimizes $\|e - \boldsymbol{H}_m \cdot y\|$

Convergence check $\|r_m = F - {}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_m\| \leq \varepsilon$

If not satisfied : $v_0 =: v_m$ go to $[\dagger]$

In the algorithm above, we have not to form the stiffness matrix, so that all the operators remain distinct at each step of the GMRES iterations. The algorithm GMRES-UDA is more efficient

than those based on the stiffness matrix such as GMRES-K. Let us note that when the GMRES-UDA has converged in $U \equiv v_m$, we also get the strain vector $\underline{\epsilon} = \hat{\mathbf{B}} \cdot U$ and the stress vector $\underline{\sigma} = \hat{\mathbf{H}} \cdot \underline{\epsilon}$, since they are intermediate computed values required by the construction of $\boldsymbol{V}_m$. We only need two auxiliary arrays to store the stress and strain vector, however their size a rather small compared to that of other arrays.

6.3. *Numerical results.*

In order to demonstrate the performance of the method proposed we apply the algorithm described above on two non-linear geomechanical problems. To check the quality of the proposed method we calculate the parameter ρ defined by:

$$\rho(N_1, N_2) = \frac{N_1\, t_1 + N_2\, t_2}{N_1\, t_1' + N_2\, t_2'},$$

where N_1 is the number of linear or linearized algebraic problems, and N_2 the number of one-GMRES($m = 1$) iterations for the whole problem. The quantities t_1 and t_2 are respectively the times allowed to one formation of $\mathbf{K}$ and GMRES($m = 1$) iteration, whereas t_1' and t_2' have the same meaning in the case of UDA.

The first problem modelled is a loading of a rigid circular fondation on an elastoplastic soil (fig. 1). From left to right, figure 1 showns the geometry of the problem, the field of vertical displacement and the horizontal one. The soil behaviour is modelled by a non-linear constitutive model[4].

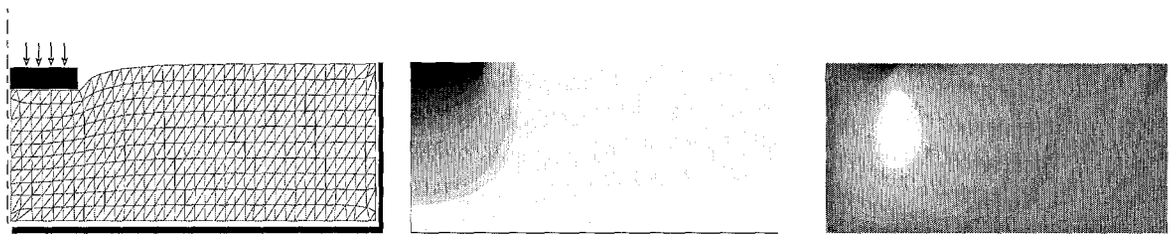

fig. 1 : Problem of the rigid circular fondation.

The number of elements is equal to 720 quadratic (T_6) or cubic (T_{10}) triangular elements. In the table 6 we recapitulate the principal features of the problem : the number of elements *nel*,the number of degrees of freedom N and the number of nonzero entries (NZK) in $\mathbf{K}$ used in a sparse matrix method.

Element	N	NZK
T_6	3066	135569
T_{10}	6758	456316

Table 6. Features of the algebraic axisymmetric problem.

The values of ρ obtained with different integration rules are shown on figures 2 and 3.

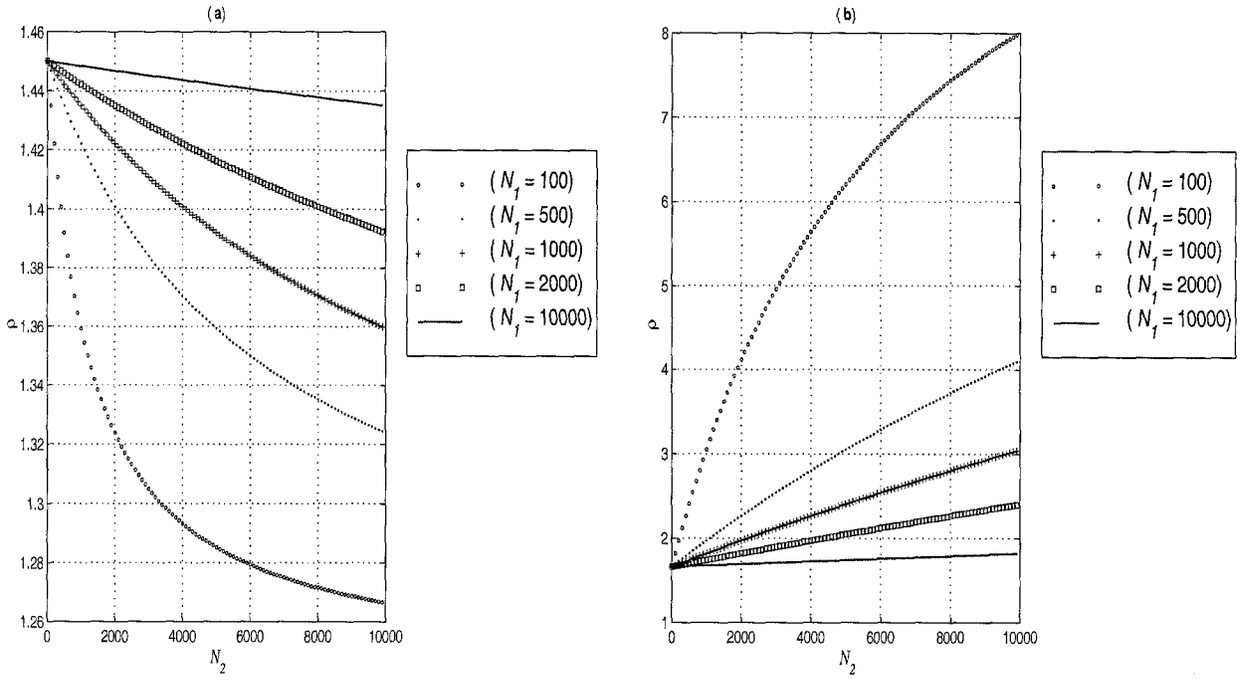

fig. 2 : Values of ρ for the quadratic element, $npi = 3(a)$ $npi = 1(b)$.

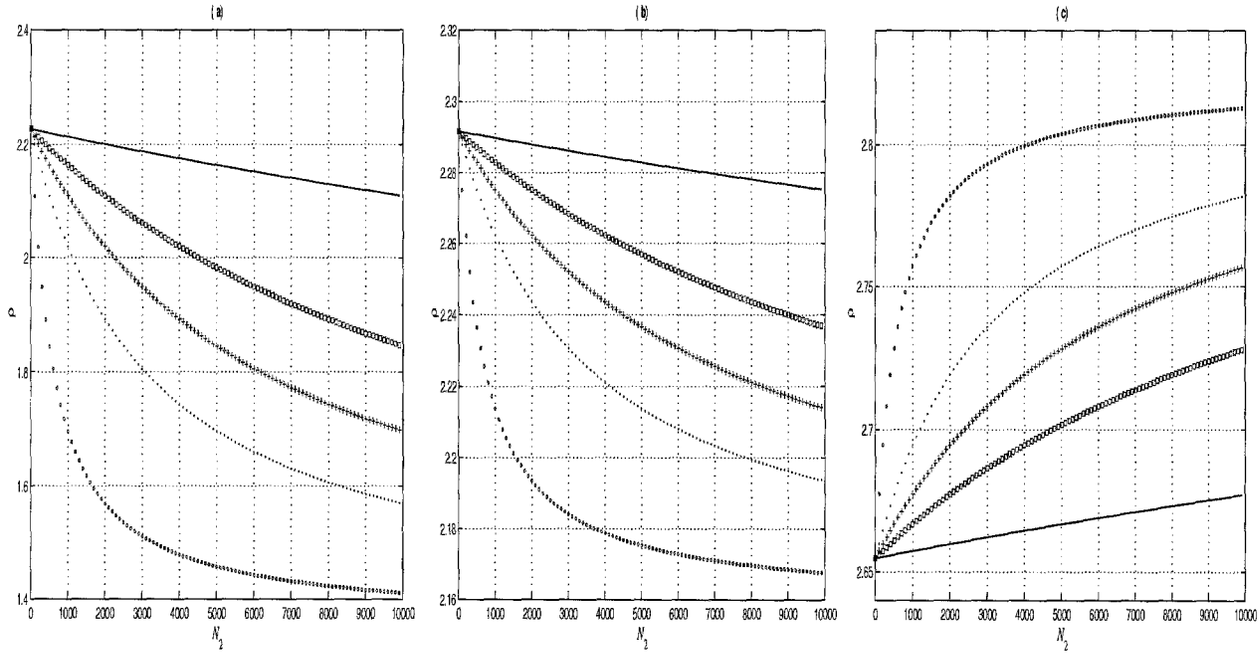

fig. 3 : Values of ρ for the cubic element, $npi = 6(a)$ $npi = 4(b)$,$npi = 3(c)$.

The second problem modelled is a soil slope loaded on its head (fig. 5). From left to right on figure 4 we have shown the geometry of the problem, the field of vertical displacement and the horizontal one.

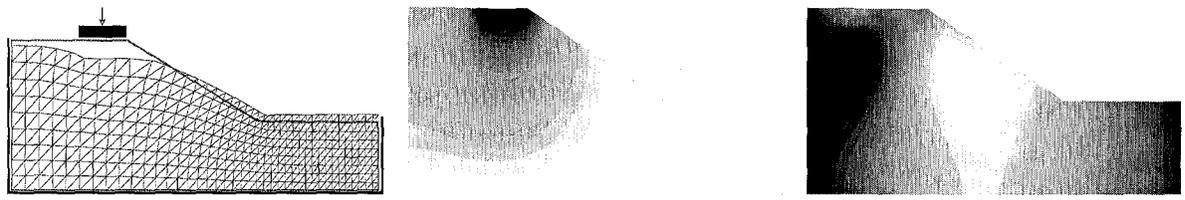

fig. 4 : Soil slope loaded on its head.

The number of elements is again equal to 720 quadratic (T_6) or cubic (T_{10}) triangular elements. In the table 8 we have recapitulated the main features of the problem.

Element	N	NZK
T_6	3078	128029
T_{10}	6776	429497

Table 7. Features of the algebraic plane strain problem.

and the results obtained with different integration rules are shown of figures 5 and 6.

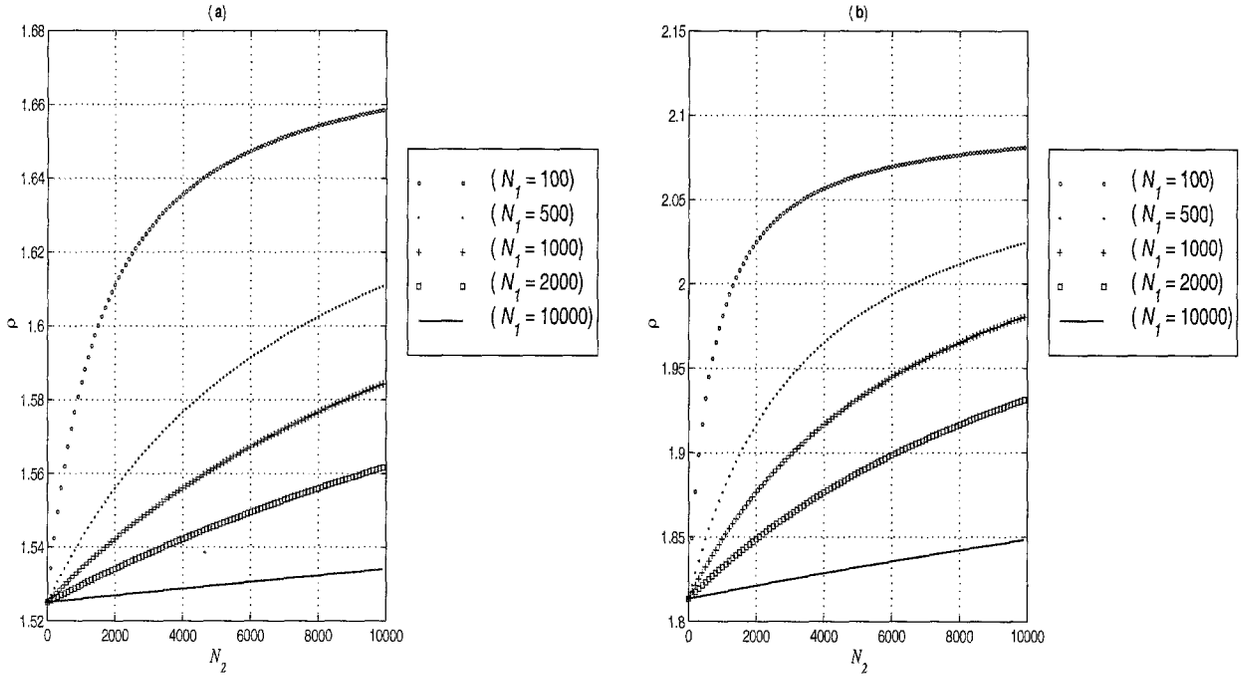

fig. 5 : Values of ρ for the quadratic element, $npi = 3(a)$ $npi = 1(b)$.

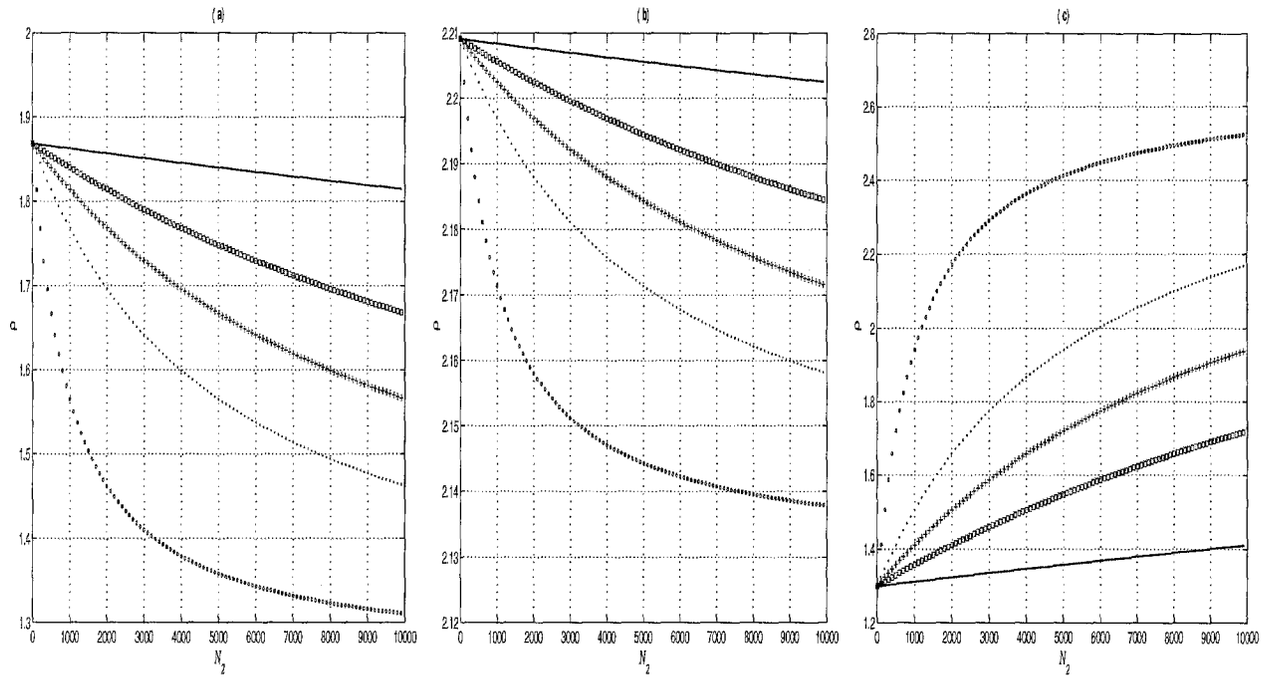

fig. 6 : Values of ρ for the cubic element, $npi = 6(a)$ $npi = 4(b)$,$npi = 3(c)$.

As we can see the values of ratio ρ are significant. They increase with the number of elements and the space-dimension of the problem. We shall now focus on the way to modify the original displacement algorithm GMRES-UDA in terms of strain or strain rate variables.

6.4. *The strain approach.*

In the previous section we have shown that the dissociated form (I) of the displacement problem is more interesting than the classical one using $\mathbf{K}$. In this section we shall show how to increase these numerical performances. It is well known that the displacement field does not belong to the set of memory parameters of the constitutive equations. This feature is very interesting in the non-linear case. And indeed, we don't need to compute the discrete displacement field of the linearized problems arising from some methods like the Newton's one.

If one is not interested in the displacement fields of these mechanical problems, one can do without computing them. If one is interested in these fields for some time values (not necessarily for all the steps of loading) one can easily recover them. Eventually, if one wants to build the displacement fields for all the steps of loading, one can do it easily without performing useless computations at each iteration of the GMRES iterative process. These features are offered by the algorithm described below.

Considering the calculation of the coefficient $h_{i,j}$ of the Hessenberg matrix we can replace the inner product in the algorithm :

$$\begin{cases} h_{i,j} & = (\mathbf{K} \cdot v_j, v_i) \\ & = {}^t v_i \cdot {}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_j \end{cases}$$

by a relation which shows off the vector of strain components :

$$h_{i,j} = {}^t\hat{\epsilon}_{v_i} \cdot \hat{\mathbf{I}}\,\hat{\mathbf{H}} \cdot \hat{\epsilon}_{v_j}, \qquad \hat{\epsilon}_{v_i} = \hat{\mathbf{B}} \cdot v_i$$

According to the orthonormality of vectors $v_i \in \mathbb{R}^N$ $i \in \{1, 2, \cdots, j\}$ we can rewrite the expression of the components of the Hessenberg matrix $h_{j+1,j}$.

$$\|h_{j+1,j}\|^2 = \|\tilde{v}_{j+1}\|^2 = \|{}^t\mathbf{B}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_j - \sum_{i=1}^{j} h_{i,j}\, v_i\|^2 \tag{24}$$

Let us define ${}^t\underline{h}_j = (h_{1,j}, h_{2,j}, \cdots, h_{j,j})$ and the basis $\boldsymbol{V}_j = [v_1, v_2, \cdots, v_j]$. Taking account of this notation the expression (24) can be written as :

$$\begin{aligned}
\|h_{j+1,j}\|^2 = \|\tilde{v}_{j+1}\|^2 &= \|{}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_j - \boldsymbol{V}_j \cdot \underline{h}_j\|^2 \\
&= \|{}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_j\|^2 + {}^t\underline{h}_j \cdot {}^t\boldsymbol{V}_j\,\boldsymbol{V}_j \cdot \underline{h}_j \\
&\quad -2\,{}^t\underline{h}_j \cdot {}^t\boldsymbol{V}_j\,{}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_j \\
&= \|{}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_j\|^2 + {}^t\underline{h}_j \cdot \underline{h}_j - 2\,{}^t\underline{h}_j \cdot \underline{h}_j \\
&= \|{}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_j\|^2 - {}^t\underline{h}_j \cdot \underline{h}_j
\end{aligned}$$

We then obtain the relation :

$$\begin{aligned}
\|h_{j+1,j}\|^2 = \|\tilde{v}_{j+1}\|^2 &= \|{}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}} \cdot v_j\|^2 - \sum_{i=1}^{j} h_{i,j}^2 \\
&= \|{}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}} \cdot \hat{\epsilon}_{v_j}\|^2 - \sum_{i=1}^{j} h_{i,j}^2
\end{aligned}$$

and the vector $\hat{\epsilon}_{v_{j+1}}$ is defined by :

$$\hat{\epsilon}_{v_{j+1}} = \hat{\mathbf{B}} \cdot v_{j+1} = \frac{1}{h_{j+1,j}}\,\hat{\mathbf{B}} \cdot \tilde{v}_{j+1}$$

After all theses algebraic transformations, the new problem does not consist in the research of the displacement vector U, but in that of the associated strain vector $\underline{\epsilon}$. In other terms

$$v_m = v_0 + \boldsymbol{V}_m \cdot y, \quad \text{with } y \in \mathbb{R}^{\,m} \text{ minimizes} \quad \|e - \boldsymbol{H}_m \cdot y\|$$

is replaced by

$$\hat{\epsilon}_{v_m} = \hat{\epsilon}_{v_0} + [\hat{\boldsymbol{\epsilon}}]_m \cdot y, \quad \text{with } y \in \mathbb{R}^{\,m} \text{ minimizes} \quad \|e - \boldsymbol{H}_m \cdot y\|,$$

$$[\hat{\boldsymbol{\epsilon}}]_m = [\hat{\epsilon}_{v_1}, \hat{\epsilon}_{v_2}, \cdots, \hat{\epsilon}_{v_m}]$$

The new algorithm expressed in term of strain can be written as follows :

Flowchart of GMRES-ϵ

$$
\left\{
\begin{array}{l}
\underline{\hat{\epsilon}}_0 \qquad \text{initial guess} \\
[\dagger]\ r_0 = F - {}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}} \cdot \underline{\hat{\epsilon}}_0 \\
\hat{\epsilon}_1 = \hat{\mathbf{B}} \cdot \dfrac{r_0}{\|r_0\|}
\end{array}
\right.
$$

$$
\left.
\begin{array}{ll}
h_{i,j} = \left(\hat{\mathbf{I}}\,\hat{\mathbf{H}} \cdot \hat{\epsilon}_j, \hat{\epsilon}_i\right) & i = \{1, 2, \cdots, j\} \\
\tilde{\hat{\epsilon}}_{j+1} = \hat{\mathbf{B}}\,{}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}} \cdot \hat{\epsilon}_j - \displaystyle\sum_{i=1}^{j} h_{i,j} \cdot \hat{\epsilon}_i & \\
h_{j+1,j} = \left(\|{}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}} \cdot \hat{\epsilon}_j\|^2 - \displaystyle\sum_{i=1}^{j} h_{i,j}^2 \right)^{1/2} & j = \{1, 2, \cdots, m\} \\
\hat{\epsilon}_{j+1} = \dfrac{\tilde{\hat{\epsilon}}_{j+1}}{h_{j+1,j}} &
\end{array}
\right\}
$$

$$
\underline{\hat{\epsilon}}_m = \underline{\hat{\epsilon}}_0 + [\hat{\boldsymbol{\epsilon}}]_m \cdot y, \quad \text{with } y \in \mathbb{R}^{\,m} \text{ minimizes } \quad \|\bar{e} - \boldsymbol{H}_m \cdot y\|
$$

Convergence check $\|r_m = F - {}^t\mathbf{B}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}} \cdot \underline{\hat{\epsilon}}_m\| \leq \varepsilon$

If not satisfied: $\underline{\hat{\epsilon}}_0 =: \underline{\hat{\epsilon}}_m$ go to $[\dagger]$

with $[\hat{\boldsymbol{\epsilon}}]_m = [\hat{\epsilon}_1, \hat{\epsilon}_2, \cdots, \hat{\epsilon}_m]$ and $\hat{\epsilon}_i = \hat{\epsilon}_{v_i}$

The difference between GMRES-UDA and GMRES-ϵ lies in the number of $\hat{\mathbf{B}}$-products which constitute the most expensive operations. In GMRES algorithm the most important computation time is that allowed to the construction of the orthonormal basis $\boldsymbol{V}_m$ of the Krylov space $\mathcal{K}_m$. A substantial reduction of this computation time leads naturally to a significant improvement of this algorithm. GMRES-UDA is more paying than GMRES-K. However, GMRES-ϵ is more paying than GMRES-UDA. And indeed the number N of $\hat{\mathbf{B}}$-products avoided is equal to :

$$N = n_{ig} \times n_{it} \times n_{ts} \tag{25}$$

where n_{ig} is the number of GMRES iterations, n_{it} the number of linearized problems (in the non-linear case) and n_{ts} the number of loading steps of the problem. For example, the table 8 give the contribution of $\hat{\mathbf{B}}$ product in percent in each $\mathbf{K} \times X$ product.

2D-Finite element	$\hat{\mathbf{B}} \times X$ (%)
Bilinear Lagrange	39.0
Linear triangle	36.4
Biquadratic Lagrange	44.4
Quadratic triangle	42.1
Bicubic Lagrange	46.7
Cubic triangle	44.9

3D-Finite element	$\hat{\mathbf{B}} \times X$ (%)
Trilinear Lagrange	40.0
Linear tetrahedron	33.3
Triquadratic Lagrange	46.6
Quadratic tetrahedron	44.4
Tricubic Lagrange	48.5
Cubic tetrahedron	44.4

Table 8. Contribution of $\hat{\mathbf{B}}$ product in each $\mathbf{K} \times X$ product.

Eventually let us note that when the GMRES-ϵ have converged in $\underline{\epsilon} \equiv \underline{\hat{\epsilon}}_m$ we obtain also the stress vector $\underline{\sigma} = \hat{\mathbf{H}} \cdot \underline{\epsilon}$, since it is an intermediate computed variable for the building of matrix $\boldsymbol{V}_m$.

If one wants to recover the displacement vector U at some given loading step, one has to solve, out of the GMRES loop and of course out of the non-linear loop, the $\mathbb{R}^{N_\sigma \times N}$ linear problem :

$$\begin{cases} \text{Find } U \in \mathbb{R}^N \text{ such that :} \\ \hat{\mathbf{B}} \cdot U = \underline{\epsilon} \end{cases} \tag{26}$$

6.4. *Preconditioning purpose*

This short section is devoted to some recommendations which can be taken into account when preconditioning of the linear or linearized problem is performed. Since the problem of an optimal preconditioning remains open, we give only some indications.

A strong restriction of the preconditioned methods is that they require to maintain the special dissociated or factorized (following Axelsson et al.[1,2]) form, at each inner (for the linearized algebraic problem) and outer (for the linearization procedure) iteration of the solving process.

The question of preconditioner is natural within iterative solving framework. They are performed in order to enhance the robustness of the iterative solvers.

The simplest preconditioned method is the diagonal one :

$$\mathbf{D} = diag\left[{}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}}\right]$$

which tansform the problem,

$$\left({}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}}\right) \cdot U = F \quad \Rightarrow \quad \mathbf{D}^{1/2}\left({}^t\hat{\mathbf{B}}\,\hat{\mathbf{I}}\,\hat{\mathbf{H}}\,\hat{\mathbf{B}}\right)\mathbf{D}^{1/2} \cdot \tilde{U} = \tilde{F}$$

with $\tilde{U} = \mathbf{D}^{1/2} \cdot U$ and $\tilde{F} = \mathbf{D}^{1/2} \cdot F$. This simple method do not need to form explicitly the stiffness matrices but require only the computing of the N diagonal entries D_{ii}.

An other direction which have to be investigated, lie in the interesting *spectral equilavence theorem* established by Axelsson et al[1]. Following Axelsson[1,2] two matrices $\mathbf{M}$ and $\mathbf{N}$ of dimension N are called spectrally equivalent if :

$$\exists c_1 > 0,\, c_2 > 0 \quad \text{such that} \quad c_1\,{}^tX \cdot \mathbf{M} \cdot X < {}^tX \cdot \mathbf{N} \cdot X < c_2\,{}^tX \cdot \mathbf{M} \cdot X \quad \forall X \in \mathbb{R}^N$$

Axelsson and Gustafsson[2] used successfully this method in a non-linear diffusion problems.

In Krylov subspace framework, more specially in GMRES subspace iterative methods, one could used the so call FGMRES referred as flexibles GMRES[23]. In this framework we dont used a unique preconditioner for the linear algebraic problem, but it perform variation with subspace iteration in order to improve the robustness of the preconditioner. Saad[23] propose to incorporate the changes in the precondtioner into the classical GMRES algorithm.

All these methods will be developped and investigated.

7. CONCLUSION

In this paper we have shown how to increase substantially the performances of the Gmres iterative algorithm when used for the numerical resolution of algebraic systems coming from the f.e.m. The theorem established shows that from an algebraic point of view the displacement one-field approach can always be considered as the dual one with subsidiary constraints. This allows

us to use algorithms which can not be employed for classical algebraic systems based on the global or elementary stiffness matrices. The numerical performances of two iterative algorithms were described and proved. One has to note that these algorithms can be easily implemented in a finite element code without requiring substantial modifications of its structure. The B-bar method[9] can be also improved thanks to our two algorithms GMRES-UDA and GMRES-ϵ. A spectral analysis can be performed by considering the dissociated form (I) of the problem, since the structure of the corresponding algebraic systems is more interesting than that of systems built with the stiffness matrix. More developments will be done, especially as concerns the coupling between the displacement one-field approach and the stress-displacement two-fields mixed one as well as the extension of the previous algorithms made to include large transformations. This last extension can be done easily without adding geometrical non-linearities to the problems modelled if one adopts a rate-type point of view involving both objective stress-rate and velocity fields.[19,20] The method will be also applied in adaptive refinement framework.

References

1. O. Axelsson and J. Maubach, On the updating and assembly of the hessian matrix in finite element methods, *Comp. Methods Appl. Mech. Eng.*, Vol. 71, 41-67 (1988).
2. O. Axelsson and I. Gustafsson, *An efficient finite element for nonlinear diffusion problems, Bulletin, Greek Mathematical Society*, Vol. 32, 45-61, 1991.
3. F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, New-York : Springer-Verlag, 1991.
4. F. Darve, 'Incrementally non-linear constitutive relationships', In, *Geomatreials constitutive equation and modelling*, (F. Darve ed.), Elsevier Applied Sciences, pp. 213-238, 1990.
5. L.C. Dutto, 'On the iterative methods for solving linear systems of equations', *Revue Européenne des éléments finis*, Vol. 2, N°. 4, 423-448 (1993).
6. M. Fortin and R. Glowinski, In *Méthodes de lagrangien augmenté: Applications à la résolution numérique de problèmes aux limites*, Paris : Dunod, 1982.
7. Y.C. Fung, *Foundations of solids mechanics.* Englewood Cliffs : Prentice-Hall, 1965.
8. G.H. Golub and C.F. Van Loan, *Matrix computations.* Baltimore : The John Hopkins University Press, 1984.
9. G.H. Golub and G.A. Meurant, *Résolution numérique des grands systèmes linéaires.* Paris : Eyrolles, 1983.
10. T.J.R. Hughes, *The finite element method: linear static and dynamic finite element analysis.* Englewood Cliffs : Prentice-Hall, 1987.
11. T.J.R. Hughes, 'Generalization of selective integration procedures to anisotropic and nonlinear media', *Int. J. Numer. Methods Eng.*, Vol. 15, 1413-1418 (1980).
12. F. Laouafa, *Analyse et développement de méthode d'éléments finis : unification et application aux lois incrémentales de type interpolation.* Thèse Doctorat: Institut National des Sciences Appliquées de Lyon et Ecole Nationale des Travaux Publics de l'Etat, 1996.
13. F. Laouafa and P. Royis, 'Eléments finis déplacements. Nouvelle forme du problème algébrique et algorithmes adaptés', *C. R. Acad. Sc. Paris*, t.325, Série. IIb 347-352 (1997).
14. D.S. Malkus and T.J.R. Hughes, 'Mixed finite element methods-reduced and selective integration techniques: a unification of concepts', *Comp. Methods Appl. Mech. Eng.*, Vol. 15, 63-81 (1978).
15. N.M. Nachtigal, S.C. Reddy and L.N. Trefethens, 'How fast are nonsymmetric matrix iterations ?', *SIAM J. Matrix Anal. Appl.*, Vol. 13, N°. 3, 778-795 (1992).
16. N.M. Nachtigal, L. Reichel and L.N. Trefethens, 'A hybrid gmres algorithm for nonsymmetric linear systems', *SIAM J. Matrix Anal. Appl.*, Vol. 13, N°. 3, 796-825 (1992).
17. T.H.H. Pian and P. Tong, 'Basis of finite element methods for solid continua', *Int. J. Numer. Methods Eng.*,Vol. 1, N°. ?, 3-28 (1969).
18. H. Royis, *Développement d'une méthode d'éléments finis mixtes pour les problèmes en grandes déformations.* Thèse Doctorat: Institut National des Sciences Appliquées de Lyon et Ecole Nationale des Travaux Publics de l'Etat, 1996.
19. P. Royis, *Formulation mathématique de loi de comportement-Modélisation numérique de problèmes aux limites en mécanique des solides déformables.* Thèse Doct. Ing.: Institut National Polytechnique de Grenoble et Ecole Nationale des Travaux Publics de l'Etat, 1986.
20. P. Royis, *Modélisation par éléments finis des géomatériaux.* Diplôme d'Habilitation à Diriger des Recherches: Institut National Polytechnique de Grenoble et Ecole Nationale des Travaux Publics de l'Etat, 1995. 140 p.
21. P. Royis, H. Royis and F. Laouafa, 'Implementation of a two-fields f.e.m. for incremental laws in large

deformations', In *Numerical Methods in Geotechnical Engineering*, Edited by Smith. Rotterdam: Balkema, p. 115-120. (1994)
22. Y. Saad and M.H. Schultz, 'GMRES: a Generalized Minimal Residual algorithm for solving nonsymmetric linear systems', *SIAM J. Sci. Statist. Comp.*, 7, 856-869 (1986).
23. Y. Saad, 'A flexible inner-outer preconditionned GMRES algorithm', *SIAM J. Sci. Comp.*, 14, 461-469 (1993).
23. F. Shakib, T.J.R. Hughes and Z. Johan, 'A multi-element group preconditioned gmres algorithm for nonsymmetric systems arising in finite element analysis', *Comp. Methods Appl. Mech. Eng.* Vol. 75, 415-456 (1989).
25. R.L. Taylor, J.C. Simo, O.C. Zienkiewicz and A.C.H. Chan, 'The patch test - a condition for assessing f.e.m. convergence', *Int. J. Numer. Methods Eng.*, Vol. 22, 39-62 (1986).
26. K. Washizu, *Variational methods in elasticity and plasticity: third edition.* Oxford : Pergamon Press, 1982.
27. O.C. Zienkiewicz, 'Mixed methods for f.e.m. and patch test, some recent developments', In *Analyse mathématique et applications*, Paris : Gauthier Villards, 473-500 (1988).
28. O.C. Zienkiewicz and S. Nakazawa, 'On variational formulation and its modification for numerical solution', *Computers and Structures*, Vol. 19, N°. 1-2, 303-313 (1984).
29. O.C. Zienkiewicz and R.L. Taylor, *The finite element method. Basic formulation and linear problems.* U.K. : Mc Graw-Hill, 1989..
30. O.C. Zienkiewicz, J.P. Vilotte, S. Toyoshima and S. Nakazawa, 'Iterative method for constrained and mixed approximation. An inexpensive improvement of f.e.m. performance', *Comp. Methods Appl. Mech. Eng.*, Vol. 51, 3-29 (1985).